UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/871,345	05/30/2001	Radu V. Ionescu	MS163104.2/40062.116USU1	9575

45979 7590 05/18/2006

PERKINS COIE LLP/MSFT
P. O. BOX 1247
SEATTLE, WA 98111-1247

EXAMINER	
RUTTEN, JAMES D	

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 05/18/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/871,345	IONESCU, RADU V.	
	Examiner	Art Unit	
	J. Derek Rutten	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 March 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-5,12,13 and 15-27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-5,12,13 and 15-27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 3/13/2006 has been entered. Claims 1-5, 12, and 16-27 have been amended, and claims 6-11 and 14 have been canceled. Claims 1-5, 12, 13, and 15-27 are pending and have been fully considered.

Response to Arguments

2. Applicant's amendments have overcome the rejections under 35 U.S.C. 101. Therefore, these rejections are withdrawn.

3. On page 7 paragraph 3 of the response filed 3/13/06, Applicant argues that claims 16-27 provide essential elements in support of the preamble. This argument is persuasive. Therefore, the prior rejection under 35 U.S.C. § 112 2nd paragraph is withdrawn.

4. On pages 9 and 10 of the response, Applicant essentially argues that Kobayashi's beans do not provide support for reviewing the operation of objects as recited in claims 1-5, 12-13, 15, and 22-27. This argument is persuasive. However, upon further consideration, a new rejection is made in view of "Experiences with cluster and class testing" by Murphy et al.

Specification

5. The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction of the following is required: Reference to the new limitation "actual parameter" (see at least claims 1, 16, and 22) was not found in the originally filed specification.

Claim Objections

6. Claim 4 is objected to because of the following informalities: The claim recites "...for a user to chose parameters...", which should likely be --for a user to choose parameters--. Appropriate correction is required.

Claim Rejections - 35 USC § 112

7. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

8. Claim 15 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claim 15 is presented as being dependent upon claim 14. However, claim 14 has been canceled, and therefore claim 15 does not carry a proper dependency. In the interest of further examination, claim 15 will be interpreted as being dependent upon claim 12.

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 1-5 and 16-26 rejected under 35 U.S.C. 103(a) as being unpatentable over prior art of record U.S. Patent 6,633,888 to Kobayashi (hereinafter referred to as "Kobayashi") in view of "Experiences with cluster and class testing" by Murphy et al. (hereinafter "Murphy").

As per claim 1, Kobayashi discloses:

A method for reviewing operation of software objects of a computer program (column 24 lines 20-62), *definitions of the software objects being stored in a library, the definitions including methods of the objects and formal parameters for the methods, the method comprising the acts of:*

instantiating <a bean> for review based on the definition of the object stored in the library; See column 22 lines 38-40:

> In step 2002, the **bean** to be tested is **loaded into the tester** described above using the visual environment add-on, also as discussed above

also column 11 lines 18-20:

> VEA 700 allows access by the visual builder 708 to beans contained in **JAR file 702**, which beans may be the beans created by the bean creator as discussed above

retrieving from an input dialog a selected method for exercising the instantiated <bean>, the input dialog identifying methods of the object based on the definition of the object stored in the library; See column 22 lines 41-42:

> Next, in step 2004, the bean is displayed in the workspace window by **selecting it from the palette**.

Note that Kobayashi's beans are proxy components for methods of an associated component object. As such, an identification of a bean is equivalent to the identification of a method of an object. Also column 22 lines 30-32:

> Testing a **composite bean** typically involves testing the operation of its **methods** and bound properties, which, as discussed above include the parameters of the methods.

obtaining an actual parameter corresponding to formal parameters specified in the definition of the object stored in the library for use in exercising the instantiated <bean> See column 22 lines 49-53:

> Further, as discussed previously, **the method parameters** of the original bean are exposed by the proxy components created from the methods of that bean. Consequently, each method can be tested fully.

Further, see column 5 lines 6-13:

> In particular, parameters associated with a method are represented by properties of the proxy component created for that method. When each proxy component is displayed on the GUI of a conventional visual builder, its properties, and consequently, **the parameters of the underlying method, are visually editable** and can be bound visually to other component properties using, for example, a mechanism in a conventional visual builder which links objects.

also column 8 lines 23-27:

> Each composite component in the application 216 can be tested within the visual builder by means of the universal transport API 206 which allows the code which implements the underlying **objects** and components 202 to be **exercised** under control of the proxy components.

exercising the instantiated <bean> with the selected method using the actual parameters so that the operation of the software <bean> is reviewed See column 22 lines 46-49:

> As previously mentioned, when the methods of proxy beans are invoked, they use the universal transport mechanism to invoke the actual component code in order to test the method as set forth in step 2008.

While disclosing the review of operation of composite beans that are based upon an associated object, Kobayashi does not expressly disclose direct review of the object

itself. However, Murphy teaches that the review of objects can be accomplished by testing class methods. See page 42 near the bottom of column 3 at bullet 5:

> A list of the classes in the cluster with an indication of whether a class is to be class tested and if so, whether the plan for testing the class exercises all methods of the class.

One of ordinary skill would have been motivated to use Murphy's teaching of class testing with Kobayashi's component objects in order to automate and reduce the cost of class testing as suggested by Murphy.

As per claim 2, the above rejection of claim 1 is incorporated. Kobayashi further discloses: *wherein the library is a type library* (column 8 lines 60-63).

As per claim 3, the above rejection of claim 1 is incorporated. Kobayashi further discloses: *wherein the act of exercising instantiates another object that may be exercised.* (column 12 line 56 – column 13 line 13).

As per claim 4, the above rejection of claim 1 is incorporated. Kobayashi further discloses: *wherein the library is a type library* (column 8 lines 60-63) *and the act of obtaining comprises: displaying an input dialog for a user to chose parameters using information stored in the type library* (column 5 lines 8-13, also FIG. 11 element 1160).

As per claim 5, the above rejection of claim 4 is incorporated. All further limitations have been addressed in the above rejection of claim 1.

In regard to claim 16, Kobayashi discloses:

A computer-readable storage medium containing instructions for controlling a computer system to test a software object (See column 23 lines 6-34), *by a method comprising:*

instantiating an object; See column 9 lines 40-44:

> The bean-based application 216 can be tested within the visual environment by using the universal transport API 206 to access the Java objects instantiated from a selected class called the "target" class in the class and bean implementations 202.

and

exercising the instantiated object by repeatedly: See column 18 lines 53-58:

> In step 1210, an operator selects components from the palette 1125 which are to comprise the ultimate composite component. In the illustrative embodiment, the **components can be selected** from the palette 1125 by holding down the "shift" key and **clicking on each component icon in turn** from the palette 1125, as desired.

Note that the language in the above passage describes a process of display and selection for plural components. This implies a repeated display and selection of components. Further, column 23 lines 3-5 describes the repeated selection and testing of components.

displaying to a user a list of methods of the object; See column 9 lines 30-39:

> In accordance with the principles of the invention, visual builder 214 includes a bean visual environment add-on 212, which **allows the proxy beans 210 to appear directly on the builder's object palette**. The proxy beans can therefore be manipulated in a conventional fashion using builder 214. Since the proxy beans include the parameters of the methods as attributes, the method parameters can be manipulated directly by changing the attributes of the proxy beans in order to generate a bean-based application 216. (emphasis added)

receiving from the user a selection of a method; See column 9 lines 30-39:

> In accordance with the principles of the invention, visual builder 214 includes a bean visual environment add-on 212, which allows the proxy beans 210 to appear directly on the builder's object palette. **The proxy beans can therefore be manipulated in a conventional fashion using builder** 214. Since the proxy beans include the parameters of the methods as attributes, the method parameters can be manipulated directly by changing the attributes of the proxy beans in order to generate a bean-based application 216. (emphasis added)

receiving from the user actual parameters for the selected method; See column 9 lines 30-39:

> In accordance with the principles of the invention, visual builder 214 includes a bean visual environment add-on 212, which allows the proxy beans 210 to appear directly on the builder's object palette. The proxy beans can therefore be manipulated in a conventional fashion using builder 214. Since the proxy beans include the parameters of the methods as attributes, **the method parameters can be manipulated directly** by changing the attributes of the proxy beans in order to generate a bean-based application 216. (emphasis added)

and

invoking the selected method of the instantiated object passing the actual parameters See column 9 lines 44-49:

> In particular, universal transport API 206 operates under control of constructor and method objects instantiated by the proxy beans 210 within bean-based application 216 to **call the appropriate constructors and methods** for the target class in the implementation code 202 as will be hereinafter explained in detail. (emphasis added)

until the methods of the instantiated object are tested. See column 9 lines 40-44:

> The bean-based application 216 can be tested within the visual environment by using the universal transport API 206 to access the Java objects instantiated from a selected class called the "target" class in the class and bean implementations 202.

In regard to claim 17, the above rejection of claim 16 is incorporated. All further limitations have been addressed in the above rejection of claim 4.

In regard to claim 18, the above rejection of claim 16 is incorporated. All further limitations have been addressed in the above rejection of claim 1.

In regard to claim 19, the above rejection of claim 16 is incorporated. Kobayashi does not expressly disclose: *repeating the instantiating and exercising for another object.* However, Kobayashi is based in an environment made up of numerous objects. See Fig. 2 element 204. It would have been obvious to one of ordinary skill in the art at the time the invention was made to repeat the instantiating and exercising for another object. One of ordinary skill would have been motivated to test more than one object so that the system could be used for testing a plurality of classes (see column 4 lines 61-63).

In regard to claim 20, the above rejection of claim 16 is incorporated. Kobayashi does not expressly disclose *logging the selection of the method and the actual parameter.* However, Murphy teaches the use of test cases, which serve as a log of selection of methods and parameters. See page 42 column 3. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Murphy's test case with Kobayashi's method and parameter selection in order to document the testing process as suggested by Murphy. Testing documentation allows comprehensive analysis of test results while informing future testing.

In regard to claim 21, the above rejection of claim 20 is incorporated. Kobayashi does not expressly disclose *logging results of the invocation.* However, Murphy teaches

logging test results. See page 43 top of column 1. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Murphy's result logging with Kobayashi's testing in order to keep a record of whether a test failed or succeeded as suggested by Murphy.

In regard to claim 22, Kobayashi discloses:

A computer-readable storage medium containing instructions for controlling a computer system to test software objects, (See column 23 lines 6-34) *each object having methods, each method having one or more formal parameters, by a method comprising:*

providing entries that specify an object, a method of the object, and an actual parameter of the method; and for each entry, instantiating <a bean> of the entry; invoking the method of the entry of the instantiated <bean> passing the actual parameter of the entry; See column 9 lines 30-39:

> In accordance with the principles of the invention, visual builder 214 includes a bean visual environment add-on 212, which allows the proxy beans 210 to appear directly on the builder's object palette. The proxy beans can therefore be manipulated in a conventional fashion using builder 214. Since the proxy beans include the parameters of the methods as attributes, the method parameters can be manipulated directly by changing the attributes of the proxy beans in order to generate a bean-based application 216.

Also see column 9 lines 44-49:

> In particular, universal transport API 206 operates under control of constructor and method objects instantiated by the proxy beans 210 within bean-based application 216 to call the appropriate constructors and methods for the target class in the implementation code 202 as will be hereinafter explained in detail.

As discussed above in connection with the rejection of claim 1, while disclosing the review of operation of composite beans that are based upon an associated object, Kobayashi does not expressly disclose direct review of the object itself. However,

Murphy teaches that the review of objects can be accomplished by testing class methods. See page 42 near the bottom of column 3 at bullet 5:

> A list of the classes in the cluster with an indication of whether a class is to be class tested and if so, whether the plan for testing the class exercises all methods of the class.

Murphy further describes instantiation of an object under review. See page 44 column 1 paragraph 3:

> The test case labeled *Creation* in Figure 5 **creates an object of class**, i.e. creates an AudioPlayer object, the class under test, and verifies that no exception occurs during the creation by specifying *noexc* (no exception).

One of ordinary skill would have been motivated to use Murphy's teaching of class testing with Kobayashi's component objects in order to automate and reduce the cost of class testing as suggested by Murphy. Also, Kobayashi does not expressly disclose *logging results of the invocation.* However, Murphy teaches logging test results. See page 43 top of column 1:

> -a section to record the results of the test

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Murphy's result logging with Kobayashi's testing in order to keep a record of whether a test failed or succeeded as suggested by Murphy.

In regard to claim 23, the above rejection of claim 22 is incorporated. Kobayashi does not expressly disclose: *wherein the entries are provided in a file.* However, Murphy teaches that entries can be provided in a file. See Figure 5.

In regard to claim 24, the above rejection of claim 22 is incorporated. All further limitations have been addressed in the above rejection of claim 1.

In regard to claim 25, the above rejection of claim 22 is incorporated. All further limitations have been addressed in the above rejection of claim 4.

In regard to claim 26, the above rejection of claim 22 is incorporated. Kobayashi further discloses: *wherein an entry includes multiple parameters.* See column 9 lines 5-19.

11. Claims 12, 13, and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kobayashi and Murphy in view of prior art of record "Comparing Microsoft Transaction Server to Enterprise JavaBeans" by Microsoft (hereinafter referred to as "MTS").

As per claim 12, Kobayashi discloses:

parsing ... object information into methods and parameters for each ... object See column 8 lines 35-37:

> Bean compiler 302 **parses** the code and extracts the relevant **methods and parameters.**

storing the methods and parameters in a library store See column 8 lines 59-63:

> As the beans are created they are inserted into a JAR file 324 by JAR File loader 322. Once all of the beans have been created, another module, the manifest file creator 308, of bean compiler 306 produces a complete **manifest file**, with an ".mf" suffix.

detecting an input selection indicating an object to be exercised See column 22 lines 41-42:

> Next, in step 2004, the bean is displayed in the workspace window by **selecting it from the palette.**

creating an instance of the object is to be exercised See column 22 lines 12-15:

> With that information specified, in step 1908, the bean tester creates the source code for the composite component, and creates manifest and JAR files for the component in step 1910.

getting the method and parameters chosen for use with the method to exercise the instance of the object See column 22 lines 49-53:

> Further, as discussed previously, **the method parameters** of the original bean are exposed by the proxy components created from the methods of that bean. Consequently, each method can be tested fully.

also column 8 lines 23-27:

> Each composite component in the application 216 can be tested within the visual builder by means of the universal transport API 206 which allows the code which implements the underlying **objects** and components 202 to be **exercised** under control of the proxy components.

invoking the method with a chosen parameters to exercise the instance of the object to be exercised See column 22 lines 46-49:

> As previously mentioned, when the methods of proxy beans are invoked, they use the universal transport mechanism to invoke the actual component code in order to test the method as set forth in step 2008.

repeating the detecting, creating, getting, <and> invoking ...for use in debugging and adjusting the operation of the ... objects See column 23 lines 3-5:

> The method then proceeds back to step 2006 where the bean is rerun to retest it.

As discussed above in connection with the rejection of claim 1, while disclosing the review of operation of composite beans that are based upon an associated object, Kobayashi does not expressly disclose direct review of the object itself. However, Murphy teaches that the review of objects can be accomplished by testing class methods. See page 42 near the bottom of column 3 at bullet 5:

> A list of the classes in the cluster with an indication of whether a class is to be class tested and if so, whether the plan for testing the class exercises all methods of the class.

Murphy further describes instantiation of an object under review. See page 44 column 1 paragraph 3:

> The test case labeled *Creation* in Figure 5 **creates an object of class**, i.e. creates an AudioPlayer object, the class under test, and verifies that no exception occurs during the creation by specifying *noexc* (no exception).

One of ordinary skill would have been motivated to use Murphy's teaching of class testing with Kobayashi's component objects in order to automate and reduce the cost of class testing as suggested by Murphy. Also, Kobayashi does not expressly disclose *logging the result of the exercising* .... However, Murphy teaches logging test results. See page 43 top of column 1:

> -a section to record the results of the test

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Murphy's result logging with Kobayashi's testing in order to keep a record of whether a test failed or succeeded as suggested by Murphy.

Also, Kobayashi does not expressly disclose COM objects. However, MTS teaches that beans (used by Kobayashi) are analogous to MTS COM objects See page 2 last paragraph:

> Each **bean** exposes its own Home interface, **analogous** to the **COM** IclassFactory interface, allowing a client to create instances of specific classes.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use MTS' teaching of COM objects in the test system of Kobayashi. One of ordinary skill would have been motivated to test MTS objects in order for the testing service to garner a large customer base since they are used at many organizations as suggested by MTS.

As per claim 13, the above rejection of claim 12 is incorporated. All further limitations have been addressed in the above rejection of claim 2.

In regard to claim 15, the above rejection of claim 12 is incorporated. Kobayashi further discloses *interpreting operations performed in exercising the instance of the object; and generating a result based upon the operations performed.* See column 22 line 66-column 23 line 3.

12. Claim 27 is rejected under 35 U.S.C. 103(a) as being unpatentable over Kobayashi and Murphy as applied to claim 22 above, and further in view of prior art of record U.S. Patent 6,519,605 to Gilgen et al. (hereinafter "Gilgen").

In regard to claim 27, the above rejection of claim 22 is incorporated. Kobayashi does not expressly disclose: *not instantiating an object that has already been instantiated.* However, in an analogous environment, Gilgen teaches that an object that has already been instantiated does not need to be instantiated. See column 15 lines 10-13. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Gilgen's teaching of object instantiation with Kobayashi's objects. One of ordinary skill would have been motivated to reuse an existing object instance in order to avoid the computational expense of creating a new one.

Conclusion

13. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. U.S. 6,407,761 to Ching et al. discloses a graphical user interface that permits the selection of objects, object methods and method parameters. See column 2 lines 49-63.

14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to J. Derek Rutten whose telephone number is (571) 272-3703. The examiner can normally be reached on T-Th 6:00-6:30, F 6:00-10:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

jdr

TUAN DAM
SUPERVISORY PATENT EXAMINER